1.

* Get the input amount.
* If the amount is greater than 0, then print as Positive.
* If the amount is equal to 0, then print as zero.
* Else the amount is negative.

2.

* Get the input passcode.
* Create a variable named digit sum.
* In the while loop condition check whether the number is not zero.
* Then calculate the remainder using the modulus operator.
* Then recalculate the number using the division operator(//).
* After the loop ends, print the sum.

3.

* Get the input number.
* Create a variable as reverse.
* In the while loop condition check whether the number is not zero.
* Then calculate the remainder using the modulus operator.
* Multiply the reverse number with 10.
* Add the remainder in the reverse.
* Then recalculate the number using the division operator(//).
* After the loop ends, print the sum.

4.

* Get input number
* Create a variable as flag
* Using for loop iterate the variable from range 1 to number + 1
* Check if any number between the range is divisible, it is divisible then increment the flag
* After the loop ends check whether the flag value is 2, then print as prime number.

5.

* Get input number
* Check it is is zero then return 1
* Create a factorial variable
* Using for loop create a range from the number to 1 with - 1 as decrement
* Multiply each value with the factorial variable.
* After the loop ends return the factorial value

6.

* Get input number
* For finding the length pass the number as string and find its length
* Create an armstrong variable..
* In the while loop condition check whether the number is not zero.
* Then calculate the remainder using the modulus operator.
* Find the power of the number with the length value and add in the armstrong variable.
* Then recalculate the number using the division operator(//).
* After the loop ends, check the amstrong number is equal to number, then print as Amstrong

7.

* Get the input password
* Collect the first and last elements of the password
* Then put it as last + password subset from index one to the previous of last element + first element, and store it in a new variable strong password.
* At last print the strong password.

8.

* Get an input number
* Create a binary variable for storing the binary format as string.
* Using a while loop checks the number, since if the number is not zero then logic inside the while loop executes.
* Inside the loop put the formula as binary = str(number % 2) + binary
* Next put this formulat number //=2
* After the loop ends print the binary value

9.

* Get the input sentence
* Split by space and save as list of strings
* Create words as string variables.
* Iterate it through a loop and check for the string size.
* If the current word size is greater than word size store that word in the word variable.
* After the loop print the work

10.

* Get two input strings
* Compare the lengths of the string if not equal, then print as not anagram.
* If the lengths are equal then compare the sorted or string1 and sorted of string2, if true then print as anagrams